ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 519.617

А.И. ДИВЕЕВ, Е.А. СОФРОНОВА

МЕТОД ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ С СЕТЕВЫМ ОПЕРАТОРОМ ДЛЯ ИДЕНТИФИКАЦИИ СИСТЕМ УПРАВЛЕНИЯ

Рассмотрена задача структурно-параметрической идентификации системы управления. Приведено ее решение по методу генетического программирования на основе сетевого оператора. Описаны свойства сетевого оператора, правила его построения и использования. Изложен принцип базисного решения, который позволяет применять генетические операции к вариациям такого решения. Показано влияние выбора базисного решения, задаваемого разработчиком на основе опыта исследования конкретной задачи, на эффективность генетического алгоритма. Приведены результаты вычислительного эксперимента решения прикладной задачи идентификации.

Ключевые слова: *идентификация систем управления, генетический алгоритм, генетическое программирование, сетевой оператор, принцип базисного решения.*

Введение. Решение задачи структурно-параметрической идентификации системы управления, как правило, проводится в два этапа: первоначально выбирается структура системы управления, а затем — значения параметров в ней. Этап выбора структуры в большинстве случаев осуществляется «вручную», т. е. в результате анализа причин наблюдаемого поведения системы. Вычислительные методы применяются только на этапе выбора значений параметров.

Метод генетического программирования [1], разработанный в 1992 г. профессором университета Стэнфорда Дж. Козой, позволяет использовать вычислительную машину для поиска алгоритма или математического выражения, т. е. с помощью вычислительной машины определить структуру системы управления. Основным элементом генетического программирования является структура данных — польская запись, — которая позволяет описывать в виде строки символов любые математические выражения или алгоритмы. Генетический алгоритм применяет генетические операции, прежде всего операцию скрещивания, к польским записям и получает новые записи, которые соответствуют новым математическим выражениям.

Для корректного использования генетического программирования требуется анализ достаточно большого множества возможных решений. При проверке каждого решения, т. е. при вычислении функции приспособленности для него или при выполнении генетических операций с ним, каждая польская запись должна быть корректно прочитана, для чего должен использоваться лексический анализатор. Работа лексического анализатора замедляет процесс вычислений. В Стэнфорде при решении задач методом генетического программирования используют мощную вычислительную машину из тысячи кластеров [5]. Другим недостатком генетического программирования, основанного на польской записи, является то, что сама запись первоначально не предусматривает включения в нее несколько раз одних и тех же аргументов, переменных или констант. Наконец, третьим недостатком генетического программирования с польской записью является то, что при ее использовании трудно ограничить область поиска. В примерах использования генетического программирования [1—4] размеры первоначальных популяций превосходили один миллион.

В генетическом программировании альтернативой польской записи является сетевой оператор. Это другая структура данных, которая позволяет описывать математические выражения с помощью ориентированного графа [6—10]. Данный граф может представляться в

вычислительной машине в виде целочисленной верхней треугольной матрицы. Сетевой оператор лишен перечисленных выше недостатков польской записи. Поэтому, если исследователю предварительно известен приблизительный вид искомого математического выражения, то он может задать некоторое базисное решение и выполнить поиск в его окрестности, что значительно сужает область поиска и уменьшает размер первоначальной популяции.

Постановка задачи. Рассмотрим систему автоматического управления, которая состоит из двух подсистем. Математическая модель одной из них известна. Математическую модель второй подсистемы требуется определить. Известны размерности каждой из подсистем и то, что модель неизвестной подсистемы может быть описана обыкновенными дифференциальными уравнениями. Необходимо по вектору управления, действующему на систему, и вектору наблюдения всей системы построить математическую модель неизвестной подсистемы.

Рассмотрим формальную постановку.

Задан вектор наблюдения

$$\mathbf{y} = \left[\mathbf{x} \vdots \mathbf{z}\right]^{\grave{o}} = \left[x_1 \ldots x_n \, z_1 \ldots z_k\right]^{\grave{o}}, \tag{1}$$

где $\mathbf{x} = \left[x_1 \ldots x_n\right]^{\grave{o}}$ — вектор состояния подсистемы с известной математической моделью, $\mathbf{x} \in \boldsymbol{R}^n$; $\mathbf{z} = \left[z_1 \ldots z_k\right]^{\grave{o}}$ — вектор состояния подсистемы с неизвестной математической моделью, $\mathbf{z} \in \boldsymbol{R}^k$.

Математическая модель известной подсистемы описывается обыкновенными дифференциальными уравнениями:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right), \tag{2}$$

где $\mathbf{u} = \left[u_1 \ldots u_m\right]^{\grave{o}}$ — вектор управления, $\mathbf{u} \in \boldsymbol{U} \subseteq \boldsymbol{R}^m$; $\boldsymbol{U}$ — ограниченное множество.

Для известного управления, действующего в течение заданного интервала времени

$$\hat{\mathbf{u}}(t), \ t \in \left[0, t_f\right] \tag{3}$$

определен вектор состояния системы

$$\hat{\mathbf{y}}(t) = \left[\hat{\mathbf{x}}(t) \vdots \hat{\mathbf{z}}(t)\right]^{\grave{o}}. \tag{4}$$

Необходимо найти математическую модель неизвестной подсистемы

$$\dot{\mathbf{z}} = \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q}\right), \tag{5}$$

где $\mathbf{q} = \left[q_1 \ldots q_p\right]^{\grave{o}}$ — вектор постоянных параметров, $\mathbf{q} \in \boldsymbol{R}^p$; $\mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q}\right)$ — неизвестная вектор-функция.

Выбор функциональной зависимости $\mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q}\right)$ и значений параметров $\mathbf{q} = \left[q_1 \ldots q_p\right]^{\grave{o}}$ должен быть таким, чтобы решение $\mathbf{y}(t)$ системы совместных дифференциальных уравнений (2) и (5) при управлении (3) с начальными условиями

$$\mathbf{y}(0) = \hat{\mathbf{y}}(0) = \left[\hat{\mathbf{x}}(0) \vdots \hat{\mathbf{z}}(0)\right]^{\grave{o}}, \tag{6}$$

минимизировало функционалы

$$J_1 = \sqrt{\int_0^{t_f} \sum_{i=1}^{n+k} \left(y_i - \hat{y}_i\right)^2}, \tag{7}$$

$$J_2 = \max_t \left\{ \left| y_i(t) - \hat{y}_i(t) \right|, \ i = \overline{1, n+k}, \ t \in \left[0, t_f\right] \right\}. \tag{8}$$

Можно считать, что функциональная зависимость выбирается из счетного множества возможных решений. Это означает, что каждой функциональной зависимости должен соответствовать целочисленный номер элемента из некоторого множества, который следует выбирать, используя методы целочисленной оптимизации. После выбора какой-либо

функциональной зависимости $\mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q})$ необходимо определить оптимальные по критериям (7) и (8) значения параметров $\mathbf{q} = \left[q_1 \dots q_p\right]^{ò}$ с помощью методов непрерывной оптимизации. Так как рассматриваются два критерия оптимизации, следовательно, задача идентификации (1)—(8) относится к классу задач многокритериальной смешанной оптимизации.

В классических методах идентификации [11, 12] функциональную зависимость $\mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q})$ разработчик выбирает на основе анализа практических свойств исследуемой системы управления. Сама идентификация предполагает только выбор значений оптимальных параметров $\mathbf{q} = \left[q_1 \dots q_p\right]^{ò}$ по методам нелинейного программирования. В последнее время для решения подобных задач идентификации используются нейронные сети [13—15], которые устанавливаются вместо неизвестной подсистемы (5) и настраиваются на основе обучающих пар (3) и (4) по критериям (7) и (8). Решение задачи идентификации на основе нейронной сети заведомо предполагает отказ разработчика от попытки найти реальную физическую закономерность, которую описывает неизвестная подсистема (5).

С помощью специальной структуры данных — сетевого оператора — строится множество функциональных зависимостей $\mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q})$:

$$\boldsymbol{G} = \left\{\mathbf{g}^i(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q}):\ i = \overline{1, M}\right\}. \qquad (9)$$

Множество (9) конечно. Для практических задач с помощью сетевого оператора несложно построить множество (9) мощностью $M \geq 10^{10}$. Далее, используя генетический алгоритм, выбираем элемент $\mathbf{g}^i(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{q})$ и находим оптимальные значения параметров $\mathbf{q}^i = \left[q_1^i \dots q_p^i\right]^{ò}$. Так как в задаче рассматриваем два критерия выбора функциональной зависимости и параметров, то решением задачи считаем построение множества Парето в пространстве функционалов (7) и (8). Рассмотрим подробнее метод решения.

Сетевой оператор. Для построения множества функциональных зависимостей (9) введем в рассмотрение несколько конечных упорядоченных множеств, из элементов которых состоит функциональная зависимость [6, 7].

Множество переменных — упорядоченное множество компонент вектора состояний объекта $\boldsymbol{V} = (v_1, \dots, v_N)$, $v_i \in \boldsymbol{R}^1$, $i = \overline{1, N}$. Множество параметров — упорядоченное множество компонент вектора параметров $\tilde{\boldsymbol{N}} = (q_1, \dots, q_p)$, $q_i \in \boldsymbol{R}^1$, $i = \overline{1, p}$. Множество унарных операций — упорядоченное множество функций или однозначных отображений, заданных на числовом множестве $\boldsymbol{O}_1 = (\rho_1(z), \rho_2(z), \dots, \rho_W(z))$, где $\rho_i(z)$: $\boldsymbol{R}^1 \to \boldsymbol{R}^1$, $\forall z \in \boldsymbol{R}^1$, $\exists y \in \boldsymbol{R}^1 \Rightarrow y = \rho_i(z)$, $i = \overline{1, W}$.

Множество бинарных операций — упорядоченное множество функций двух аргументов или однозначных отображений декартового произведения пары одинаковых числовых множеств в одно такое же числовое множество $\boldsymbol{O}_2 = (\chi_0(z', z''), \chi_1(z', z''), \dots, \chi_{V-1}(z', z''))$, где $\chi_i(z', z'')$: $\boldsymbol{R}^1 \times \boldsymbol{R}^1 = \boldsymbol{R}^2 \to \boldsymbol{R}^1$, $\forall z', z'' \in \boldsymbol{R}^1$, $\exists y \in \boldsymbol{R}^1 \Rightarrow y = \chi_i(z', z'')$, $i = \overline{0, V-1}$.

Сетевой оператор — это ориентированный граф, обладающий следующими свойствами:

а) в графе отсутствуют циклы;

б) к любому узлу, который не является источником, есть хотя бы один путь от узла-источника;

в) от любого узла, который не является стоком, есть хотя бы один путь до узла-стока;

г) каждому узлу-источнику соответствует элемент из множества переменных $\boldsymbol{V}$ или из множества параметров $\boldsymbol{C}$;

д) каждому узлу соответствует бинарная операция из множества $\boldsymbol{O}_2$ бинарных операций;

е) каждой дуге графа соответствует унарная операция из множества $\boldsymbol{O}_1$ унарных операций.

Вычисления по сетевому оператору включают три этапа.

На первом этапе находим узел, который имеет выходящие дуги, но не имеет ни одной входящей дуги. Для этой цели берем любой узел и перемещаемся от него к следующему по одной из входящих дуг. Продолжаем перемещение до тех пор, пока не найдем узел, у которого нет входящих дуг. Первоначально таким узлом является узел-источник. На втором этапе выполняем унарную и бинарную операции. Унарная операция соответствует дуге, выходящей из найденного узла. В качестве аргумента унарной операции используем значение, которое хранится в найденном узле. Бинарная операция соответствует узлу, в который дуга входит. В качестве первого аргумента бинарной операции используем либо единицу, либо результат последнего вычисления, который хранится в данном узле. В качестве второго аргумента используем результат вычисления унарной операции. На третьем этапе исключаем узел и дугу из графа. Найденный узел исключаем в том случае, если этот узел не является узлом-стоком и не имеет ни одной выходящей дуги. Дугу исключаем во всех случаях, если операция, которая соответствует данной дуге, выполнена.

Этапы вычислений выполняем до тех пор, пока в сетевом операторе не останутся только узлы-стоки, в которых хранятся результаты вычислений.

При вычислении следует соблюдать правило, при котором выполнение унарной операции согласно выходящей из узла дуге, осуществляется только в том случае, если в узел, из которого данная дуга выходит, не входит ни одна дуга.

Рассмотрим пример. Пусть задан сетевой оператор (рис. 1). В узлах-источниках расположены значения из множества переменных $\boldsymbol{V} = (t)$ и множества параметров $\tilde{\boldsymbol{N}} = (\alpha, \beta, \omega, \varphi)$. В остальных узлах расположены номера унарных операций $\boldsymbol{O}_2 = \left(\chi_0(z', z'') = z' + z'', \chi_1(z', z'') = z'z''\right)$, а на узлах расположены цифры, соответствующие номерам унарных операций $\boldsymbol{O}_1 = \left(\rho_1(z) = z, \rho_2(z) = \sin z,\right.$ $\left.\rho_3(z) = -z, \rho_4(z) = e^z\right)$. Тогда, используя правила вычисления результата по сетевому оператору, получаем математическое выражение:

$$y = \alpha e^{-\beta t} \sin(\omega t + \varphi).$$

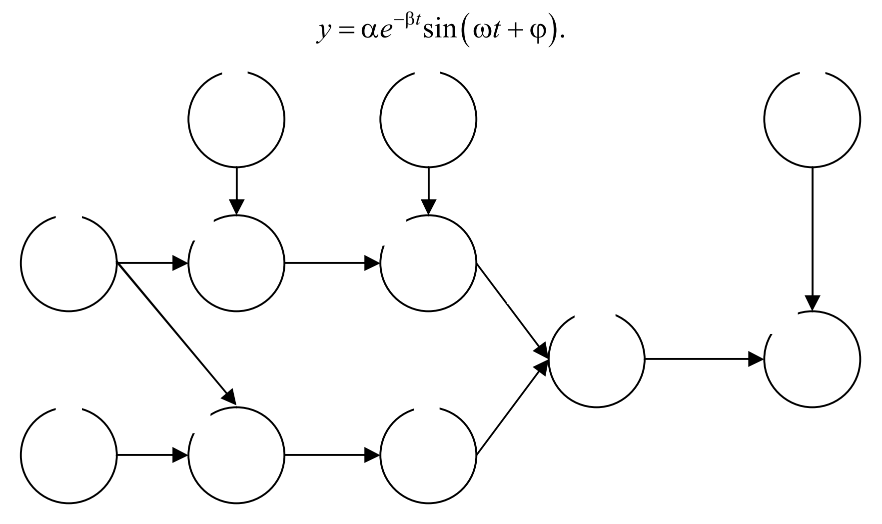


Рис. 1. Пример сетевого оператора

Матрица сетевого оператора — это целочисленная матрица, на диагонали которой расположены номера бинарных операций, а остальные элементы либо нули, либо номера унарных операций, причем при замене диагональных элементов на нули, а ненулевых недиагональных элементов на единицы получаем матрицу смежности графа сети, удовлетворяющую условиям «а—в» определения сетевого оператора.

Для сетевого оператора (см. рис. 1) матрица имеет вид:

$$\mathbf{\Psi}=\begin{bmatrix} 0&0&0&0&0&1&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&2&0\\ 0&0&0&0&0&0&0&1&3&0&0\\ 0&0&0&0&0&0&0&0&0&4&0\\ 0&0&0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}.$$

Для вычислений результата математического выражения по матрице сетевого оператора необходимо определить дополнительные целочисленные векторы:

— вектор номеров узлов для входных переменных $\mathbf{b}=\left[b_1 \dots b_n\right]^{\text{ò}}$, где b_i — номер узла-источника в сетевом операторе, с которым связана переменная x_i, $i=\overline{1,\, n}$;

— вектор номеров узлов для параметров $\mathbf{s}=\left[s_1 \dots s_l\right]^{\text{ò}}$, где s_i — номер узла-источника в сетевом операторе, с которым связан параметр q_i, $i=\overline{1,\, l}$;

— вектор номеров узлов для выходных переменных $\mathbf{d}=\left[d_1 \dots d_m\right]^{\text{ò}}$, где d_i — номер узла сетевого оператора, который соответствует выходной переменной y_i, $i=\overline{1,\, m}$. Условие $m>1$ соответствует тому, что сетевой оператор описывает не одну формулу.

Введем вектор узлов $\mathbf{z}=\left[z_1 \dots z_L\right]^{\text{ò}}$, размерность L которого равна количеству узлов сетевого оператора.

Теорема. Пусть задана формула, состоящая из бинарных и унарных операций, и для формул построен сетевой оператор, который описывается следующими элементами: матрицей сетевого оператора $\mathbf{\Psi}=\left[\psi_{ij}\right]$, $i,j=\overline{1,\, L}$, вектором номеров узлов для входных переменных $\mathbf{b}=\left[b_1 \dots b_n\right]^{\text{ò}}$, вектором номеров узлов для параметров $\mathbf{s}=\left[s_1 \dots s_l\right]^{\text{ò}}$ и вектором номеров узлов для выходных переменных $\mathbf{d}=\left[d_1 \dots d_m\right]^{\text{ò}}$. Тогда этих элементов достаточно для вычисления формулы.

Доказательство. Зададим начальные значения вектора узлов

$$z_i^{(0)}=\begin{cases} v_k, & \text{åñëè } i=b_k,\ k=\overline{1,n},\\ c_j, & \text{åñëè } i=s_j,\ j=\overline{1,l},\\ e_{\psi_{ii}}, & \text{åñëè } i\notin\left\{b_1,\dots,b_n\right\}\cup\left\{s_1,\dots,s_l\right\} \end{cases},\ \chi_{\psi_{ii}}\left(e_{\psi_{ii}},z\right)=z,$$

где $i=\overline{1,\, L}$; $e_{\psi_{ii}}$ — единичный элемент для бинарной операции $\chi_{\psi_{ii}}$.

Просматриваем элементы матрицы Ψ, находящиеся над главной диагональю. Если $\psi_{ij} \neq 0$, то $z_j^{(i)} = \chi_{\psi_{jj}}\left(z_j^{(i-1)}, \rho_{\psi_{ij}}\left(z_i^{(i-1)}\right)\right)$, $i = \overline{1, L-1}$, $j = \overline{i+1, L}$.

Покажем, что $z_i^{(L-1)}$, $i = d_k$, $k = \overline{1, m}$ содержит значения вычислений по формулам, которые описывает сетевой оператор. Для этой цели необходимо показать, что выполнены все операции графических записей и не нарушен порядок раскрытия скобок.

Для ненулевого элемента ψ_{ij} матрицы Ψ сетевого оператора выполняются унарная операция, соответствующая дуге (i, j), и бинарная операция, соответствующая узлу j, следовательно, все операции сетевого оператора будут выполнены, кроме узла 1. Согласно топологической сортировке этот узел может быть только источником и с ним не связана бинарная операция.

Пусть в формуле аргументами какой-либо бинарной операции являются унарные операции $\chi_k\left(\rho_p(z'), \rho_r(z'')\right)$. В соответствии с топологической сортировкой номер узла j, с которым связана бинарная операция $k = \psi_{jj}$, должен быть больше номеров узлов i и k, от которых идут дуги к узлу j соответствующих унарных операций $p = \psi_{ij}$, $r = \psi_{kj}$, поэтому $j > i$ и $j > k$. Таким образом, унарные операции будут выполнены раньше, чем бинарная.

Пусть в формуле аргументом унарной операции является бинарная операция $\rho_k\left(\chi_p(z', z'')\right)$, $k = \psi_{ij}$, $p = \psi_{ii}$, $i < j$. Следовательно, бинарная операция будет выполнена раньше унарной. Согласно алгоритму бинарная операция $\chi_{\psi_{ii}}(z', z'')$ будет выполняться для всех ненулевых элементов $\psi_{ki} \neq 0$ матрицы сетевого оператора, находящихся в столбце i, $k < i$.

Переход к следующей строке возможен только, если выполнены все операции в вышерасположенных строках. Операция $\rho_{\psi_{ij}}\left(\chi_r(z', z'')\right)$ будет выполнена при условии, если вычислены все унарные операции $\rho_{\psi_{ki}}\left(\chi_{\psi_{kk}}(z', z'')\right)$, $k < i$, которые соответствуют всем входящим в узел i дугам. Следовательно, при выполнении алгоритма 1 соблюдается правило последовательности вычисления унарных операций, порядок раскрытия скобок не нарушается.

Принцип базисного решения. Для построения алгоритма идентификации системы управления используем принцип базисного решения [6, 7]. Принцип заключается в том, что при решении оптимизационных задач первоначально строим базисное решение, которое является одним из допустимых решений, затем определяем малые вариации базисного решения и создаем алгоритм поиска, который ищет оптимальное решение на множестве малых вариаций.

Определим следующие малые вариации сетевого оператора: 0 — замена унарной операции на дуге графа; 1 — замена бинарной операции в узле графа; 2 — добавление в граф дуги с унарной операцией; 3 — удаление унарной операции с дугой графа.

Для описания любой вариации достаточно использовать целочисленный вектор из четырех компонент $\mathbf{w} = \left[w_1\ w_2\ w_3\ w_4\right]^{\text{ò}}$. Первая компонента указывает на номер вариации, вторая — на номер узла, из которого выходит дуга, третья — на номер узла, в который входит дуга, четвертая — на номер унарной или бинарной операции. Все вариации сетевого оператора можно также выполнять на матрице сетевого оператора $\Psi = \left[\psi_{ij}\right]$, $i, j = \overline{1, L}$. Тогда вторая и третья компоненты вектора вариаций указывают соответственно на номера строки и столбца матрицы сетевого оператора.

Пусть задана матрица сетевого оператора $\mathbf{\Psi}^0 = \left[\psi_{ij}^0\right]$, $i, j = \overline{1, L}$ и вектор вариаций $\mathbf{w}^1 = \left[w_1^1 \; w_2^1 \; w_3^1 \; w_4^1\right]^{\text{ò}}$. В результате действия вектора вариаций $\mathbf{w}^1$ на матрицу сетевого оператора $\mathbf{\Psi}^0$ получаем новую матрицу сетевого оператора $\mathbf{\Psi}^1 = \left[\psi_{ij}^1\right]$, $i, j = \overline{1, L}$, $\mathbf{\Psi}^1 = \mathbf{w}^1 \circ \mathbf{\Psi}^0$.

Каждая вариация изменяет один элемент матрицы $\mathbf{\Psi}$ сетевого оператора. Если заданы две матрицы сетевых операторов $\mathbf{\Psi}^0 = \left[\psi_{ij}^0\right]$, $i, j = \overline{1, L}$ и $\tilde{\mathbf{\Psi}} = \left[\tilde{\psi}_{ij}\right]$, $i, j = \overline{1, L}$ одинаковой размерности L, то количество несовпадающих между ними элементов определяет количество вариаций, которые необходимо выполнить, чтобы из одной матрицы получить другую.

Для двух несовпадающих матриц сетевых операторов всегда можно найти вариации, применение которых к одной матрице позволяет получить другую, причем количество вариаций равно расстоянию между матрицами $\tilde{\mathbf{\Psi}} = \mathbf{w}^d \circ \ldots \circ \mathbf{w}^1 \circ \mathbf{\Psi}^0$.

Генетический алгоритм. Для поиска решения задачи идентификации (1)—(8) используем генетический алгоритм, построенный на основе принципа базисного решения.

Первоначально задаем базисное решение задачи, которое может представлять собой линейную систему по наблюдаемым координатам пространства состояний или любую другую систему, полученную из опыта разработчика системы управления. Базисное решение описываем матрицей сетевого оператора $\mathbf{\Psi}^0 = \left[\psi_{ij}^0\right]$, $i, j = \overline{1, L}$.

В качестве возможного решения используем упорядоченное множество векторов вариаций и битовую строку, описывающую в коде Грея вектор параметров искомого математического выражения $\boldsymbol{W}^i = \left(\mathbf{w}^{i1}, \ldots, \mathbf{w}^{il}\right)$, $\mathbf{s}^i = \left[s_1^i \ldots s_{p(c+d)}^i\right]^{\text{ò}}$, $i = \overline{1, H}$, где H — количество возможных решений или размерность популяции; p — количество параметров в искомом математическом выражении; c — число бит целой части; d — число бит дробной части.

Для перевода значений параметров из кода Грея $\mathbf{s}^i$ используем следующие соотношения:

$$q_k^i = \sum_{j=1}^{c+d} 2^{c-j} b_{j+(k-1)(c+d)}^i, \; k = \overline{1, p},$$

где $b_j^i = \begin{cases} s_j^i, \; \text{åñëè} \; (j-1) \bmod (c+d) = 0, \\ s_j^i \oplus b_{j-1}^i, \; \text{èí à÷å.} \end{cases}$

Здесь $j = \overline{1, p(c+d)}$.

Для каждой хромосомы $\left(\boldsymbol{W}^i, \mathbf{s}^i\right)$, $1 \le i \le H$ определяем значения функционалов. Для этой цели по структурной части $\boldsymbol{W}^i = \left(\mathbf{w}^{i1}, \ldots, \mathbf{w}^{il}\right)$ хромосомы и базисной матрице $\mathbf{\Psi}^0$ строим матрицу сетевого оператора $\mathbf{\Psi}^i = \mathbf{w}^{i1} \circ \ldots \circ \mathbf{w}^{il} \circ \mathbf{\Psi}^0$.

Рассматриваемая задача идентификации использует два функционала (7) и (8), поэтому решением задачи является множество Парето, для построения которого в генетическом алгоритме применяем ранговый критерий. Для каждого возможного решения в популяции вычисляем значение ранга доминирования

$$\Lambda^k = \sum_{h=1}^{H} \lambda_{hk}, \qquad (10)$$

где $\lambda_{hk} = \begin{cases} 1, \; \text{åñëè} \; \mathbf{J}^h \le \mathbf{J}^k, \\ 0, \; \text{èí à÷å.} \end{cases}$

Здесь $\mathbf{J}^k = \left[J_1^k \; J_2^k\right]^{\text{ò}}$.

Решения, имеющие $\Lambda^k = 0$, принадлежат множеству Парето $\boldsymbol{P}_c = \left\{ \left(\boldsymbol{W}^k, \mathbf{s}^k \right), 1 \le k \le H: \Lambda^k = 0 \right\}$. Для репродукции хромосом в популяции случайно отбираем пару $\left(\boldsymbol{W}^{h_1}, \mathbf{s}^{h_1} \right)$ и $\left(\boldsymbol{W}^{h_2}, \mathbf{s}^{h_2} \right)$ и определяем вероятность скрещивания по условию

$$p_s = \max\left\{ \frac{1+\gamma\Lambda^{h_1}}{1+\Lambda^{h_1}}, \frac{1+\gamma\Lambda^{h_2}}{1+\Lambda^{h_2}} \right\}, \tag{11}$$

где $\gamma \in [0, 1)$.

При скрещивании получаем четыре новых возможных решения $\left(\boldsymbol{W}^{H+1}, \mathbf{s}^{H+1} \right)$, $\left(\boldsymbol{W}^{H+2}, \mathbf{s}^{H+2} \right)$, $\left(\boldsymbol{W}^{H+3}, \mathbf{s}^{H+3} \right)$, $\left(\boldsymbol{W}^{H+4}, \mathbf{s}^{H+4} \right)$. В решениях $\left(\boldsymbol{W}^{H+1}, \mathbf{s}^{H+1} \right)$, $\left(\boldsymbol{W}^{H+2}, \mathbf{s}^{H+2} \right)$ сохраняется структура родителей, но изменяется параметрическая часть. В решениях $\left(\boldsymbol{W}^{H+3}, \mathbf{s}^{H+3} \right)$, $\left(\boldsymbol{W}^{H+4}, \mathbf{s}^{H+4} \right)$ изменяются и структурная и параметрическая части.

Для новых хромосом выполняем операцию мутации с заданной вероятностью p_m. Алгоритм завершает свою работу после выполнения заданного числа поколений. Результатом работы является условное множество Парето, построенное на конечной популяции возможных решений. Алгоритм также может завершить свою работу, если условное множество Парето не изменяется на протяжении заданного количества поколений.

Пример. Рассмотрим нелинейную систему управления четвертого порядка. Известная и неизвестная подсистемы в математической модели системы имеют порядок $n = k = 2$. Модель известной подсистемы представлена в виде

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = x_1 - x_1 z_1 + u_1.$$

Наблюдается вектор состояния $\mathbf{y} = \left[x_1\ x_2\ z_1\ z_2 \right]^{\text{т}}$ на интервале времени $t_f = 10$ с. Вектор управления имеет две компоненты $\mathbf{u} = \left[u_1\ u_2 \right]^{\text{т}}$. Идентификация проводится в 21 точке.

Входной вектор управления имеет следующий вид:

$$u_1 = \sin\left(\frac{t_f}{2\pi} t \right), \quad u_2 = \cos\left(\frac{t_f}{2\pi} t \right).$$

При моделировании были выбраны следующие параметры генетического алгоритма: размерность популяции — 1024, число поколений — 64, число скрещиваемых пар в одном поколении — 256, длина хромосомы — 8, число бит под целую часть — 2, число бит под дробную часть — 14, размерность матрицы сетевого оператора 20, число элитарных хромосом — 16, вероятность мутации — 0,8, параметр скрещивания — 0,4.

Заданы следующие конструктивные множества: переменных $\boldsymbol{V} = \left(x_1, x_2, z_1, z_2, u_1, u_2 \right)$, параметров $\boldsymbol{C} = \left(q_1, q_2 \right)$, унарных операций $\boldsymbol{O}_1 = \left(z, -z, z^{-1}, \frac{1-e^{-z}}{1+e^{-z}}, \sin z, \cos z, \operatorname{arctg} z, \sqrt[3]{z} \right)$ и бинарных операций $\boldsymbol{O}_2 = \left(z' + z'', z'z'' \right)$.

В результате получена следующая модель неизвестной подсистемы управления:

$$\dot{z}_1 = q_2 z_2,$$
$$\dot{z}_2 = q_1 z_1 x_1 \cos z_2 + u_2 + x_1 - q_1 z_1 - q_2 z_2,$$

где $q_1 = 1,2091064453125$, $q_2 = 0,6346435546875$.

Для данной модели $J_1 = 2,983$, $J_2 = 1,0004$.

На рис. 2 приведены результаты моделирования системы управления с полученной моделью неизвестной подсистемы. Точками обозначены наблюдаемые значения переменных.

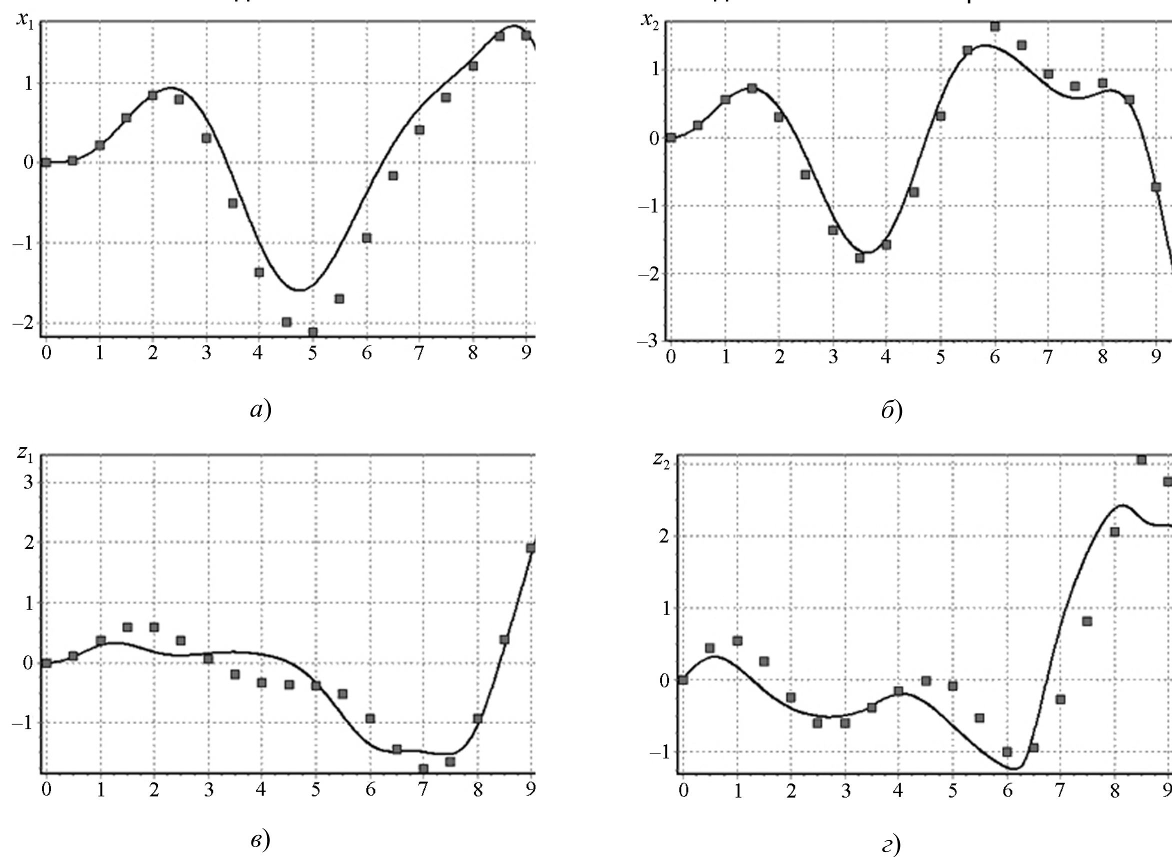


Рис. 2. Наблюдаемые и полученные значения

Заключение. В результате использования метода генетического программирования с сетевым оператором удалось идентифицировать полностью правую часть подсистемы системы управления. Полученная математическая модель представляет собой нелинейные уравнения. Решение всей системы удовлетворяет наблюдаемым значениям компонент.

Работа выполнена по теме гранта РФФИ № 10-08-00618-а.

Библиографический список

1. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. 1992. — 840 p.

2. Koza J.R. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press. 1994.

3. Koza J.R., Bennet F.H.B., Andre D., Keane M.A. Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufman Publisher. 1999.

4. Koza J.R., Keane M.A., Streeter M.J., Midlovec W., Yu J., Lanza G. Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers. 2003.

5. http: //www.genetic-programming.org.

6. Дивеев А.И., Софронова Е.А. Задача структурного синтеза системы автоматического управления // Вестн. Рос. ун-та дружбы народов. Сер.: Инженерные исследования. — 2007. — № 1. — С. 48—58.

7. Дивеев А.И., Софронова Е.А. Метод генетического программирования для автоматического подбора формул в задаче структурного синтеза системы управления // Тр. ин-та систем. анализа РАН. Динамика неоднородных систем / под ред. чл.-кор. РАН Ю.С. Попкова. — М. : ИСА РАН : КомКнига, 2006. — Вып. 10(1). — С. 14—26.

8. Дивеев А.И., Софронова Е.А. Метод сетевого оператора в задачах управления // Вестн. Рос. ун-та дружбы народов. Сер.: Инженерные исследования. — 2007. — № 4. — С.107—118.

9. Дивеев А.И., Софронова Е.А. Генетический алгоритм для многокритериального структурно-параметрического синтеза // Там же. — С. 126—131.

10. Дивеев А.И., Софронова Е.А. Метод сетевого оператора для идентификации систем управления // Там же. — 2008. — № 4. — С. 78—85.

11. Дейч А.М. Методы идентификации динамических объектов. — М. : Энергия, 1989. — 240 с.

12. Эйксхофф П. Основы идентификации систем управления. — М. : Мир, 1979.

13. Каллан Р. Основные концепции нейронных сетей. — М. : Вильямс, 2003.

14. Тюкин И.Ю., Терехов В.А. Адаптация в нелинейных динамических системах. — М. : Изд-во ЛКИ, 2008. — 384 с.

15. Demuth H., Beale M., Hagan M. Neyral Network ToolBox^{TM} 6: User’s Guide. 2008. MathWork, Inc.

References

1. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. 1992. — 840 p.

2. Koza J.R. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press. 1994.

3. Koza J.R., Bennet F.H.B., Andre D., Keane M.A. Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufman Publisher. 1999.

4. Koza J.R., Keane M.A., Streeter M.J., Midlovec W., Yu J., Lanza G. Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers. 2003.

5. http: //www.genetic-programming.org.

6. Diveev A.I., Sofronova E.A. Zadacha strukturnogo sinteza sistemy avtomaticheskogo upravleniya // Vestn. Ros. un-ta drujby narodov. Ser.: Injenernye issledovaniya. — 2007. — № 1. — S. 48—58. — in Russian.

7. Diveev A.I., Sofronova E.A. Metod geneticheskogo programmirovaniya dlya avtomaticheskogo podbora formul v zadache strukturnogo sinteza sistemy upravleniya // Tr. in-ta sistem. analiza RAN. Dinamika neodnorodnyh sistem / pod red. chl.-kor. RAN Yu.S. Popkova. — M. : ISA RAN : KomKniga, 2006. — Vyp. 10(1). — S. 14—26. — in Russian.

8. Diveev A.I., Sofronova E.A. Metod setevogo operatora v zadachah upravleniya // Vestn. Ros. un-ta drujby narodov. Ser.: Injenernye issledovaniya. — 2007. — № 4. — S. 107—118. — in Russian.

9. Diveev A.I., Sofronova E.A. Geneticheskii algoritm dlya mnogokriterial'nogo strukturno-parametricheskogo sinteza // Tam je. — S. 126—131. — in Russian.

10. Diveev A.I., Sofronova E.A. Metod setevogo operatora dlya identifikacii sistem upravleniya // Tam je. — 2008. — № 4. — S. 78—85. — in Russian.

11. Deich A.M. Metody identifikacii dinamicheskih ob'ektov. — M. : Energiya, 1989. — 240 s.

12. Eikshoff P. Osnovy identifikacii sistem upravleniya. — M. : Mir, 1979. — in Russian.

13. Kallan R. Osnovnye koncepcii neironnyh setei. — M. : Vil'yams, 2003. — in Russian.

14. Tyukin I.Yu., Terehov V.A. Adaptaciya v nelineinyh dinamicheskih sistemah. — M. : Izd-vo LKI, 2008. — 384 s. — in Russian.

15. Demuth H., Beale M., Hagan M. Neyral Network ToolBoxTM 6: User's Guide. 2008. MathWork, Inc.

Материал поступил в редакцию 10.04.10.

A.I. DIVEYEV, E.A. SOFRONOVA

GENETIC PROGRAMMING METHOD FOR CONTROL SYSTEMS IDENTIFICATION

A problem of structural-parametric control systems identification is considered. The method of genetic programming with the network operator is used. Network operator properties, rules of its creation and application are described. The basic solution approach is used for the genetic algorithm construction. It permits to apply genetic operations to the basic solution variations. The basic solution is set by a developer on the ground of his experience of concrete problem research. The selection of the basis solution influences on effectiveness of genetic algorithm. The results of computational experiment of the problem of identification solution are presented.
Key words: *control systems identification, genetic algorithm, genetic programming, network operator, basic solution approach.*

ДИВЕЕВ Асхат Ибрагимович (р. 1954), заведующий сектором Проблем кибернетики Учреждения Российской академии наук Вычислительного центра им. А.А. Дородницина РАН, доктор технических наук (2001), профессор (2009). Окончил МВТУ им. Н.Э. Баумана (1980).
Область научных интересов — теория управления, эффективные вычислительные алгоритмы и дискретная оптимизация.
Автор более 130 научных работ.
idiveev@mail.ru

СОФРОНОВА Елена Анатольевна, доцент кафедры «Кибернетика и мехатроника» Российского университета дружбы народов, кандидат технических наук (2008). Окончила РУДН (2006).
Область научных интересов — теория управления, эффективные вычислительные алгоритмы, синтез и идентификация систем управления.
Автор более 30 научных работ.
sofronova_ea@mail.ru

Askhat I. Diveyev (1954), Head of the Cybernetics Problems Sector, Institution of the Computer Centre Russian Academy of Science after A.A. Dorodnitsin. Ph.D. in Science (2001), Professor (2009). He graduated from Bauman Moscow Highest Technical School (1980).
Research interests — control theory, effective computational algorithms and discrete optimization.
Author of more than 130 scientific publications.
aidiveev@mail.ru

Elena A. Sofronova, Associate Professor of the Cybernetics and Mechatronics Department, Peoples' Friendship University of Russia. Candidate of Science in Engineering (2008). She graduated from Peoples' Friendship University of Russia (2006).
Research interests — control theory, effective computational algorithms, synthesis and identification of control systems.
Author of more than 30 scientific publications.
sofronova_ea@mail.ru